**PRACTICAL 1**

**Select and Write down the problem statement for a real time system of relevance.**

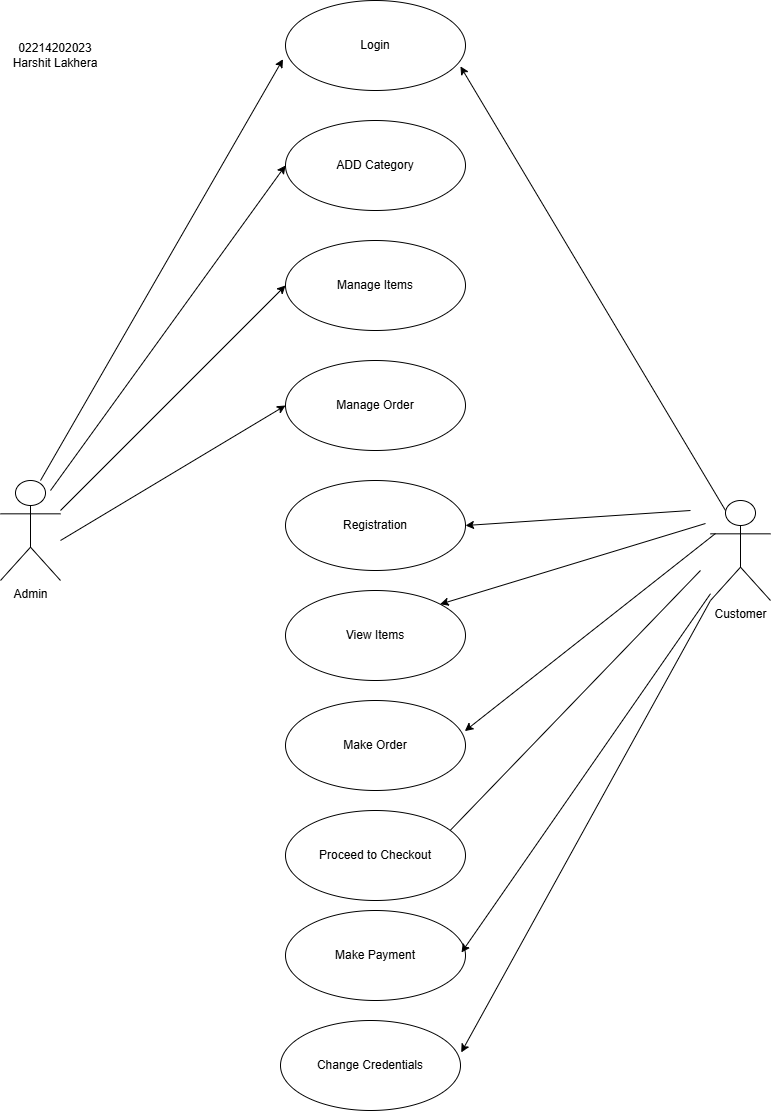
**Problem statement:**

This study will support and manage the development of a real-time e-commerce platform for Amazon aimed at enhancing the online shopping experience for customers. A successful e-commerce platform relies on providing accurate, real-time product information and managing inventory effectively. However, maintaining this accuracy and efficiency, especially during peak traffic periods (sales, holidays), presents significant challenges. These challenges impact customer experience, operational efficiency, and ultimately, revenue. Specifically, the platform faces the following issues:

* I**naccurate Real-time Inventory:** Discrepancies between displayed inventory and actual stock levels lead to overselling (selling items that are out of stock) and stockouts (items being unavailable despite appearing in stock). Overselling results in order cancellations, customer dissatisfaction, and logistical complications. Stockouts lead to lost sales opportunities and frustrated customers.
* **Scalability Challenges During Peak Traffic:** The system struggles to handle the surge in requests during peak periods. This results in slow loading times, errors, and potentially system crashes, further exacerbating inventory inaccuracies and hindering the shopping experience.
* **Complex Product Information Management:** Managing and updating product information (descriptions, images, pricing, variations) across a large catalog is a complex and time-consuming process. Inaccurate or outdated information leads to customer confusion, returns, and negative reviews.
* **Integration with Legacy Systems:** The e-commerce platform may need to integrate with existing legacy systems (e.g., warehouse management, ERP). These integrations can be complex and prone to errors, particularly when handling real-time data updates.
* **Lack of Real-time Insights:** The platform lacks real-time insights into inventory levels, sales trends, and customer behavior. This makes it difficult to optimize inventory management, personalize recommendations, and proactively address potential issues.

**PRACTICAL 2**

**To perform the user’s view analysis for the suggested system:   
Use case diagram**

****

**Admin Use Cases:**

**Login**

* **Description:** Allows the administrator to access the system's administrative functions.
* **Actors:** Admin
* **Precondition:** The system is running.
* **Basic Flow:**
  1. Admin navigates to the admin login page.
  2. Admin enters valid username and password.
  3. System verifies credentials.
  4. System grants access to the admin dashboard.
* **Alternative Flow:**
  1. Invalid credentials: System displays an error message.
  2. Account locked: System prompts for password reset.
* **Postcondition:** Admin is authenticated and logged in.
* **Extension Points:** None

**Add Product Category**

* **Description:** Enables the administrator to add new product categories to the system.
* **Actors:** Admin
* **Precondition:** Admin is logged in.
* **Basic Flow:**
  1. Admin navigates to the "Add Category" section.
  2. Admin enters the category name and description.
  3. System validates input.
  4. System creates the new category.
* **Alternative Flow:** Invalid input: System displays an error message.
* **Postcondition:** New category is created and stored.
* **Extension Points:** None

**Manage Items**

* **Description:** Allows the administrator to manage (add, edit, delete) products.
* **Actors:** Admin
* **Precondition:** Admin is logged in.
* **Basic Flow:**
  1. Admin navigates to the "Manage Products" section.
  2. Admin selects action (add, edit, delete).
  3. Admin enters/modifies product details.
  4. System validates input.
  5. System updates product information.
* **Alternative Flow:** Invalid input: System displays an error message.
* **Postcondition:** Product information is updated.
* **Extension Points:** Integration with inventory management system.

**Manage Orders**

* **Description:** Enables the administrator to view, process, and manage customer orders.
* **Actors:** Admin
* **Precondition:** Admin is logged in.
* **Basic Flow:**
  1. Admin navigates to the "Manage Orders" section.
  2. Admin selects an order to view/manage.
  3. Admin can update order status, process refunds, etc.
* **Alternative Flow:** Order not found: System displays an error message.
* **Postcondition:** Order status is updated.
* **Extension Points:** Integration with shipping and payment systems.

**Customer Use Cases:**

**Login**

* **Description:** Customer accesses their account on the e-commerce platform.
* **Actors:** Customer
* **Precondition:** Registered account, login functionality available.
* **Basic Flow:** Customer enters credentials, system verifies, grants access**.**
* **Alternative Flows:** Invalid credentials, account not found, account inactive/suspended, forgot password.
* **Postcondition: Successful login:** Customer is authenticated and session is created. Failed login: Error message displayed.
* **Extension Points:** "Remember Me" option.

**Registration**

* **Description:** Allows new customers to register an account.
* **Actors:** Customer
* **Precondition:** Customer is on the registration page.
* **Basic Flow:**
  1. Customer fills out the registration form.
  2. System validates input.
  3. System creates the new customer account.
* **Alternative Flow:** Invalid input: System displays an error message.
* **Postcondition:** New customer account is created.
* **Extension Points:** Email verification.

**View Items**

* **Description:** Allows the customer to browse and view available products.
* **Actors:** Customer
* **Precondition:** Customer is on the website.
* **Basic Flow:**
  1. Customer navigates through categories or uses search.
  2. System displays relevant products.
* **Alternative Flow:** No products found: System displays a message.
* **Postcondition:** Product listings are displayed.
* **Extension Points:** Product filtering and sorting.

**Make Order**

* **Description:** Enables the customer to add a product to their shopping cart.
* **Actors:** Customer
* **Precondition:** Customer is viewing product details.
* **Basic Flow:**
  1. Customer clicks "Add to Cart" button.
  2. System adds the product to the cart.
* **Alternative Flow:** Product out of stock: System displays a message.
* **Postcondition:** Product is added to the cart.
* **Extension Points:** Option to add multiple quantities.

**Proceed to Checkout**

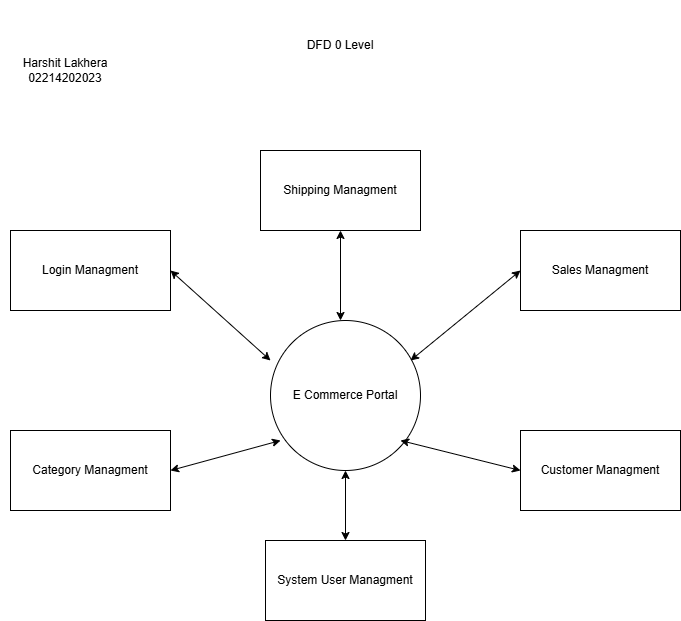
* **Description:** Initiates the checkout process.
* **Actors:** Customer
* **Precondition:** Customer has items in their cart.
* **Basic Flow:** Customer clicks "Proceed to Checkout."
* **Alternative Flow:** None
* **Postcondition:** Checkout process is initiated.
* **Extension Points:** Guest checkout option.

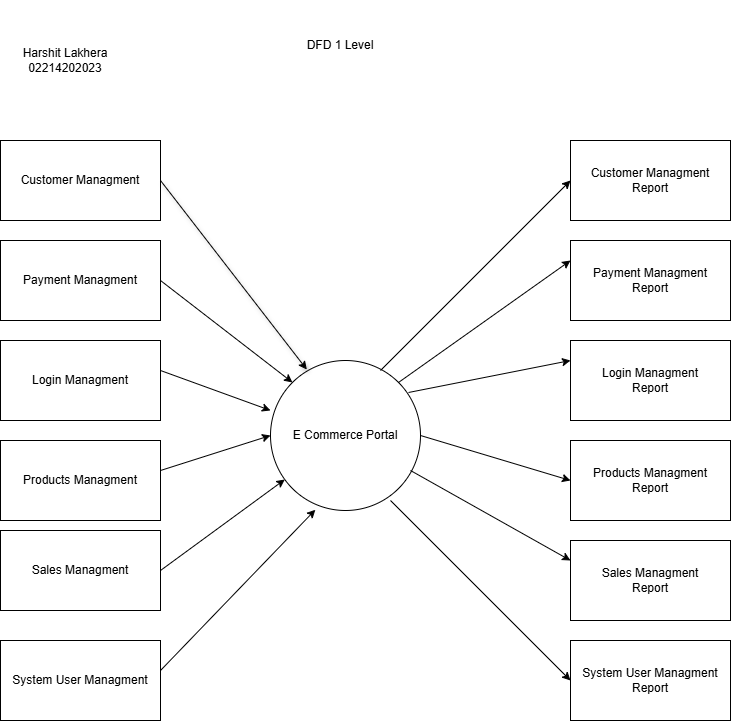
**Make Payment**

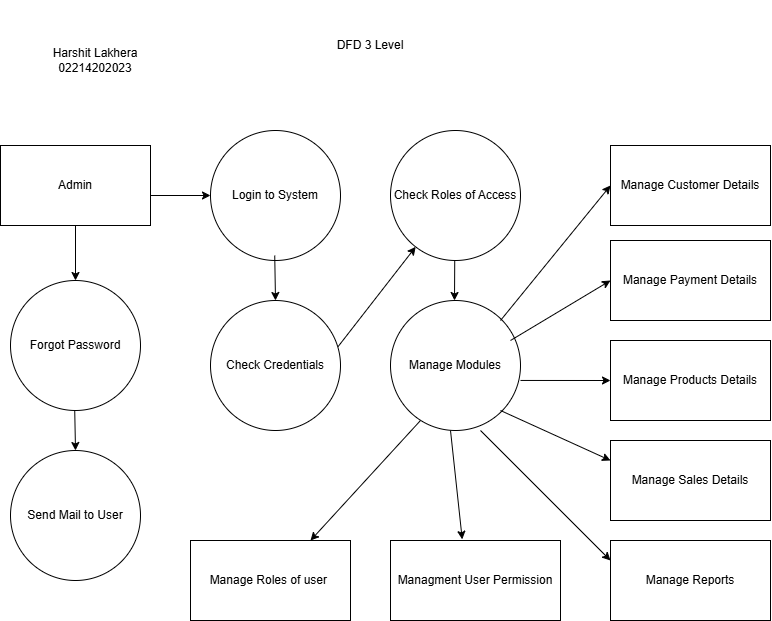
* **Description:** Allows the customer to make payment for their order.
* **Actors:** Customer
* **Precondition:** Customer has proceeded to checkout.
* **Basic Flow:**
  1. Customer selects a payment method.
  2. Customer provides payment information.
  3. System processes the payment.
* **Alternative Flow:** Payment failed: System displays an error message.
* **Postcondition:** Payment is processed.
* **Extension Points:** Integration with various payment gateways.

**PRACTICAL 3**

**To create the function oriented diagram: Data Flow Diagram (DFD)**

****

****

****

**PRACTICAL 4**

**To draw the structural view diagram for the system: Class diagram**

**Class Diagram:**

A class diagram is a type of static structure diagram in UML (Unified Modeling Language) that represents the structure and relationships of classes, interfaces, and their associations in a system. It provides an overview of the classes and their attributes, methods, and associations in the system.

**Components of a Class Diagram:**

**Class**: A class represents a blueprint or template for creating objects.

**Attributes**: Attributes represent the characteristics or properties of a class. Methods: Methods represent the behaviors or operations that a class can perform. Associations: Associations represent the relationships between classes.

**Multiplicity**: Multiplicity defines the number of instances or objects that participate in an association.

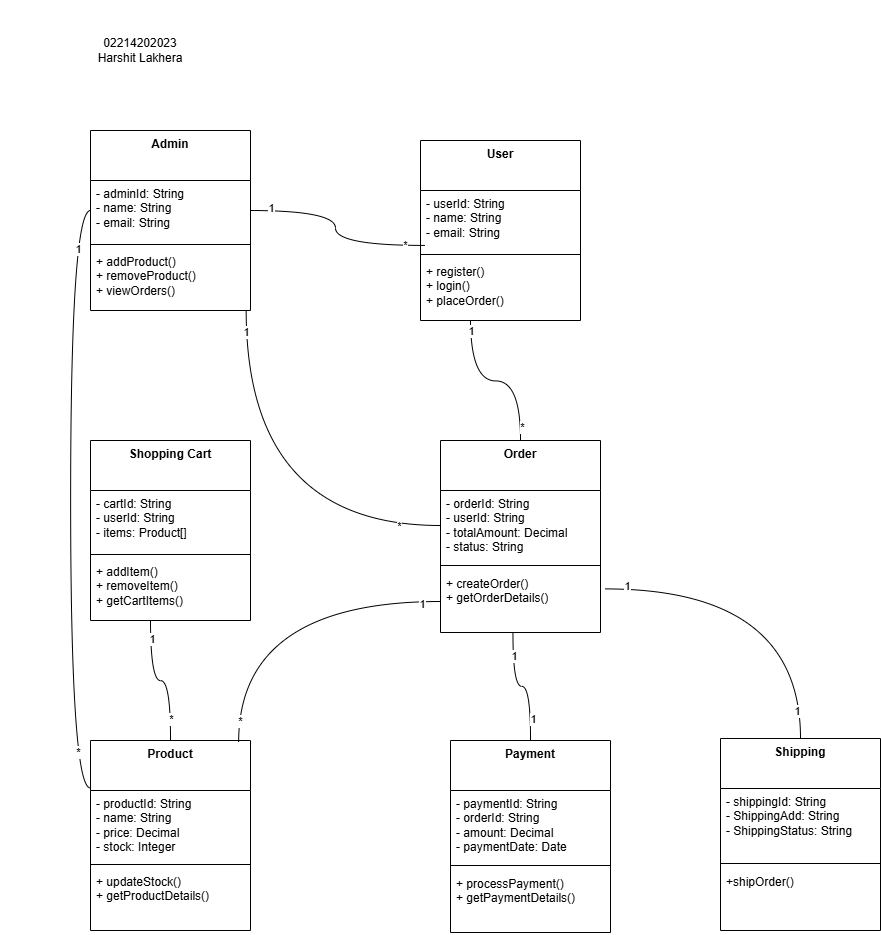
**Inheritance**: Inheritance represents the "is-a" relationship between classes.

**Interfaces**: Interfaces define a contract or set of methods that a class must implement.

**Dependency**: Dependency represents a relationship where one class depends on another class.

**Aggregation and Composition**: Aggregation and composition represent the whole-part relationships between classes.

**Stereotypes and Notes**: Stereotypes and notes provide additional information or annotations to enhance the understanding of the class diagram.

****

**PRACTICAL 5**

**To perform the behavioral view diagram for the suggested system**

**: Sequence diagram**

**SEQUENCE DIAGRAM:**

The sequence diagram represents the flow of messages in the system and is also termed as an event diagram.

**COMPONENTS OF SEQUENCE DIAGRAM:**

**1. Actors:** Actors represent the external entities that interact with the system.

**2. Objects:** Objects are the specific instances of classes or components within the system**.**

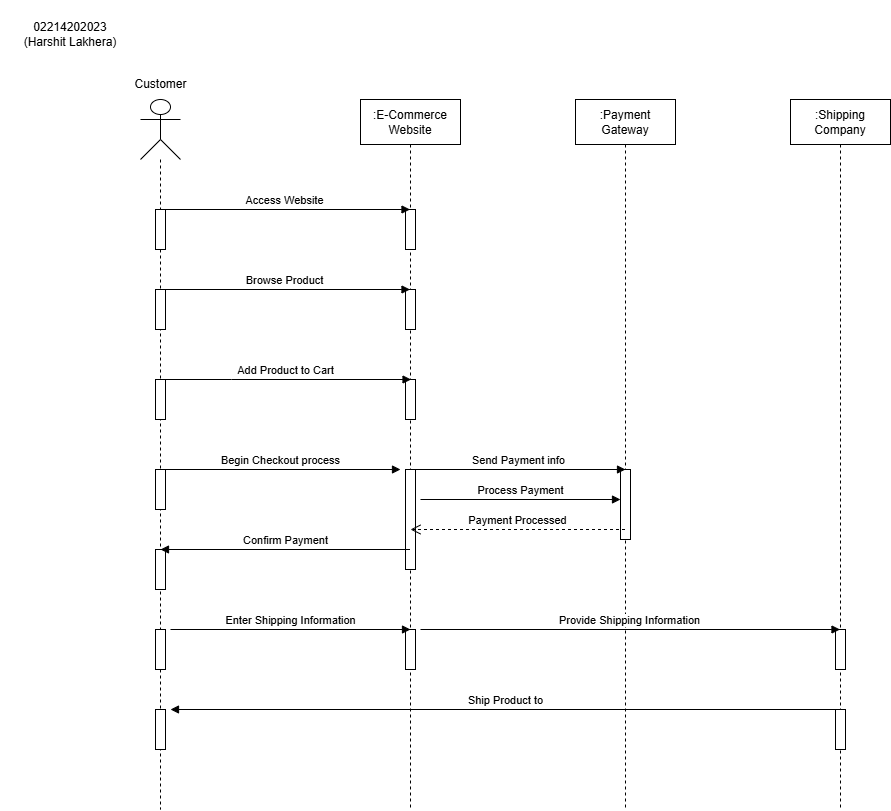
**3. Lifelines:** Lifelines represent the lifespan of an object during the execution of a sequence diagram.

**4. Messages:** Messages depict the communication or method calls between objects.

**5. Activation Boxes:** Activation boxes, also known as execution specifications, represent the period of time during which an object is executing a method or performing an action.

**6. Combined Fragments:** Combined fragments are used to represent conditional or looping behavior within a sequence diagram.

**7. Notes:** Notes provide additional explanations, comments, or annotations to enhance the understanding of the sequence diagram.

****